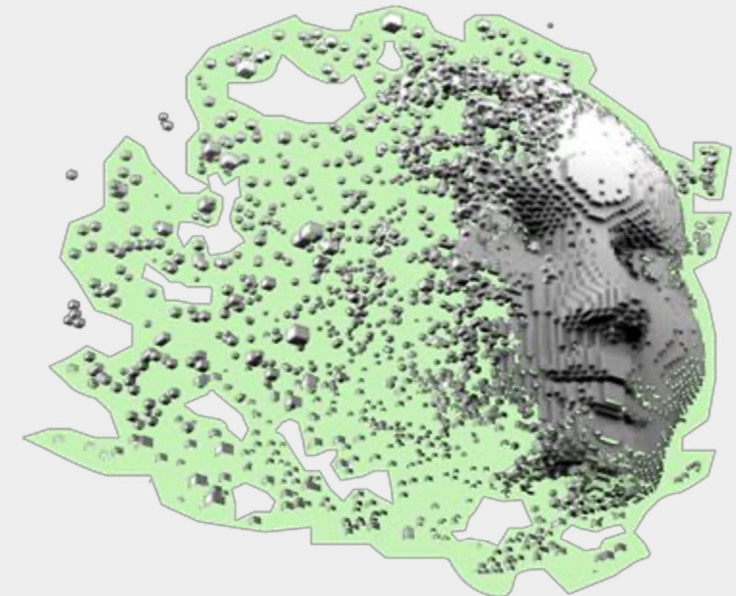
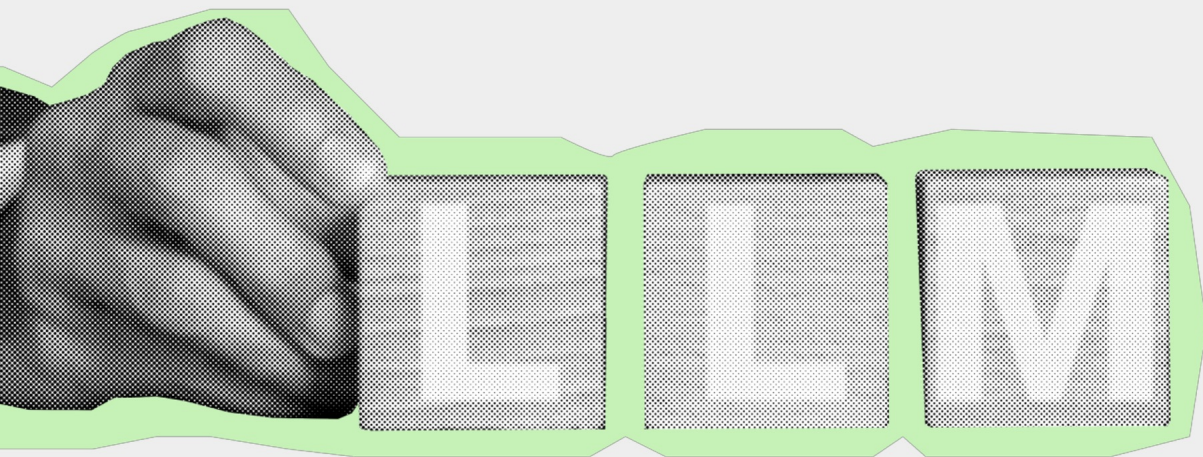


Knowledge Bases for AI Agents

03.09.2026 / 16:00 – 18:00





AI:AT × Dobriy AI

AI:AT Webinar

Knowledge Bases for AI Agents

From hyperlinked notes to structured, queryable and trustworthy knowledge.

Daniel Dobriy

Dobriy AI · Bilateral AI Cluster of Excellence · WU Vienna

03 September 2026

16:00–18:00 CEST · Online

Learning outcomes

What we will learn & build, **step by step.**

01	02	03	04
 Linked knowledge base Notes, links, tags and structured metadata.	 Document-RAG agent Retrieve relevant passages and cite their sources.	 Knowledge Graph Explicit entities, relationships, identity and provenance.	 Graph-RAG agent SPARQL for structure, documents for descriptive context.



Daniel Dobriy

Daniel is a member of the **Bilateral AI Cluster of Excellence** and an AI researcher & lecturer at **WU Vienna**. He is active in the **W3C RDF & SPARQL Working Groups** and serves as industry leader in the **GOBLIN COST Action**. He is the Founder & Managing Director of **Dobriy AI**, a member of **AI:AT**, building the European Google for Data and the infiniscalable data infrastructure for Hybrid AI. Daniel is also an editor of the **Journal of Web Semantics** and a programme committee member for leading conferences in **AI, the Web and information retrieval**.

Dobriy AI

dobriy.ai
Company



dobriy.net
Personal page



RAGE-KG 2026 — International academic workshop

Retrieval-Augmented Generation Enabled by Knowledge Graphs

The leading venue for **KG-based RAG**.

Research momentum

The community is
expanding rapidly.

Submissions increased sharply in 2026,
reflecting growing research interest in KG-
based RAG.

rage-kg.org 

ISWC 2026 · Bari, Italy · 25-26 October

Don't ask your model to become the database.



LANGUAGE MODEL

Generates language

- Interprets questions ("transformer")
- Does limited reasoning ("chain-of-thought")
- Plans actions (ReAct)

PROBABILISTIC · GENERAL · "FAST THINKING"

≠



KNOWLEDGE BASE

Maintains facts

Stores current, domain-specific and inspectable information.

GOVERNED · QUERYABLE · "SLOW THINKING"

Use language models as **natural language interface**, knowledge bases for **evolving facts/memory**, and tools for **controlled access** between them.

Running case

European AI Ecosystem.

Three question complexity levels.

01



Direct lookup

When is the Knowledge Bases for AI Agents webinar, and who can attend?

02



Bounded multi-source

Can an SME use AI:AT's EuroHPC services to fine-tune a model, test agent inference and host the resulting application in production?

03



Multi-hop + aggregation

Can an SME go from no AI-agent experience to a source-grounded workflow on European HPC—and what path gets them there?

The complexity of questions you want to handle determines how structured your knowledge base needs to be.

Hyperlinked notes · 01

Start with a **note**.

• • • agentic-ai-bootcamp.md

European Agentic AI Bootcamp

A hands-on training in building, connecting and querying knowledge bases for reliable, up-to-date AI agents.

- Dates: 20–22 July 2026
- Format: online
- Participants: 250+
- Support team: 30
- Hands-on compute: Athena

Source: AI:AT programme and event report



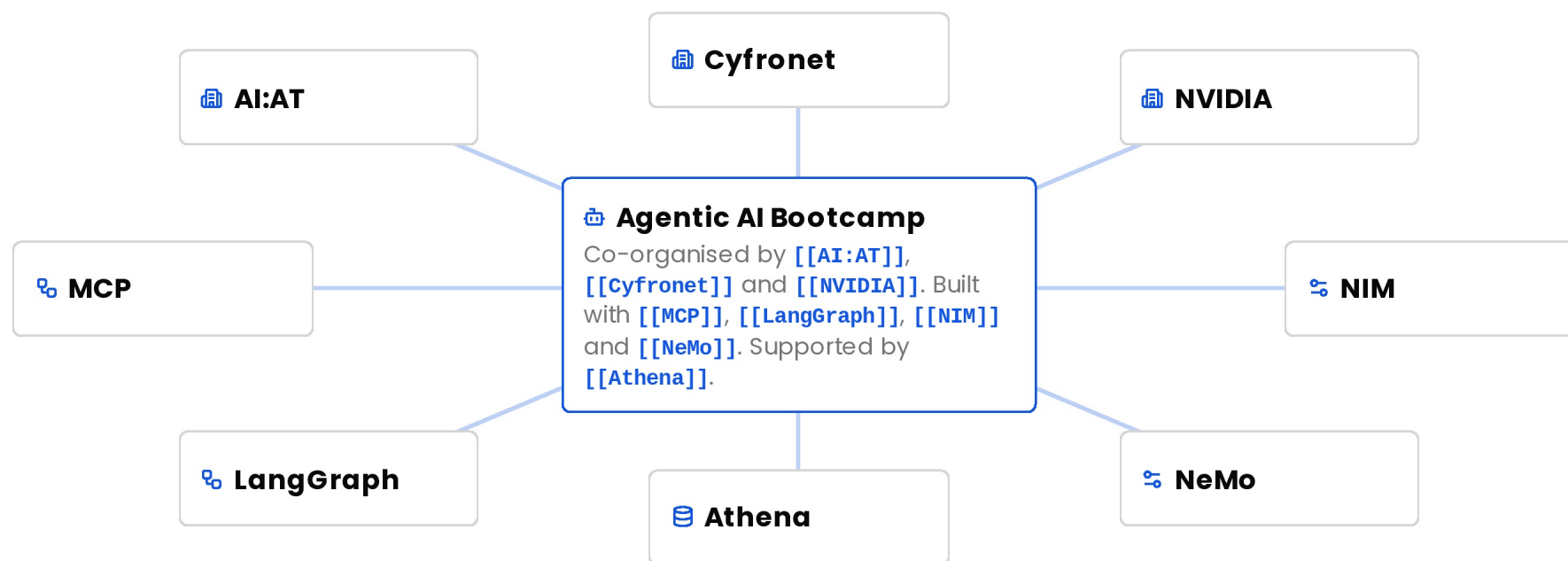
A practical unit of external memory

Readable by a human, searchable by semantic similarity.

-
- One topic per note
 - Stable links to related pages
 - Citations beside each claim

Hyperlinked notes · 02

Connect notes with **links**.



↔ A [\[\[wikilink\]\]](#) makes a relationship navigable, but its meaning still lives in the prose and is not formal.

Hyperlinked notes • 03

From a folder to a **wiki**.

📁 European AI Ecosystem

📁 Events

📄 European Agentic AI Bootcamp.md

📁 Organisations

📄 AIAT.md

📄 Cyfronet.md

📄 HLRS.md

📁 Technologies

📁 Infrastructure



Links

Connect pages and support navigation.



Tags

Create lightweight categories and filters.



Frontmatter

Add dates, types, status and identifiers.



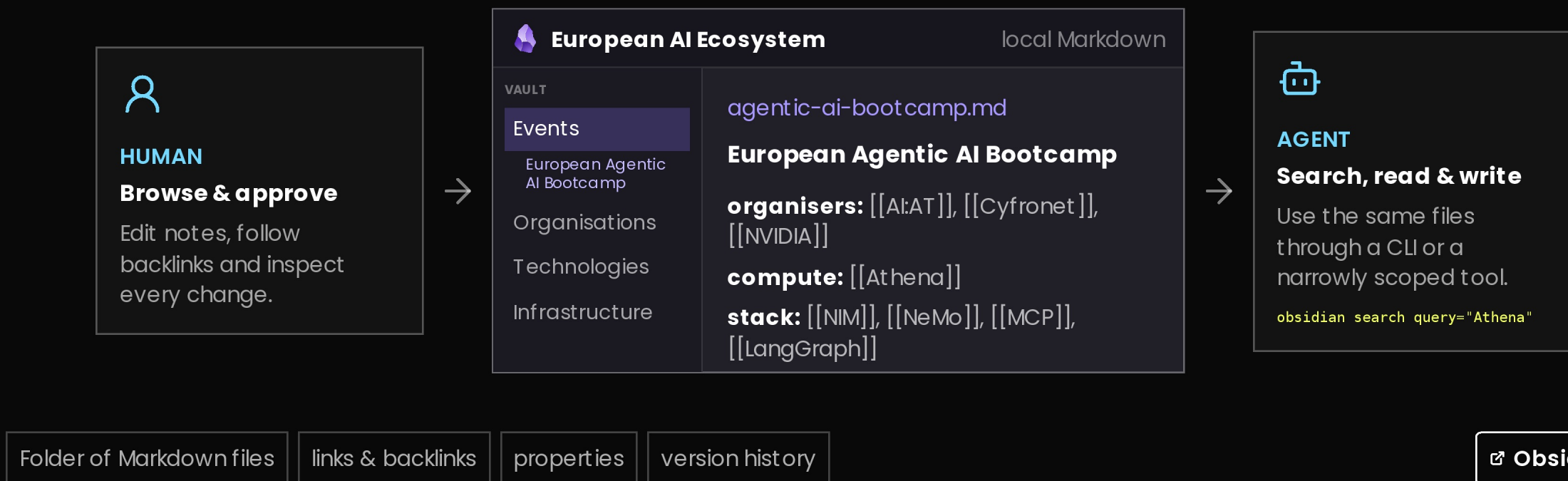
Templates

Keep repeated page structures consistent.

A wiki is already a human-maintained knowledge system. Its structure is useful, but mostly **implicit**.

Obsidian + agents

A personal knowledge base for humans and agents.



Agent access modalities

How agents can work with an Obsidian vault.

01



DIRECT FILE ACCESS

Markdown & folders

`*.md + directories`

The agent reads and edits ordinary vault files through filesystem tools.

- + Simplest local setup
- Link updates and write conflicts

02



OFFICIAL INTERFACE

Obsidian CLI

`obsidian search ...`

The agent calls Obsidian's own search, link, property and file commands.

- + App-aware automation
- Obsidian must be running

03



TOOL PROTOCOL

MCP server

`read_note · search`

The server exposes selected vault operations as tools to an MCP-capable agent.

- + Explicit, client-neutral tools
- Permissions vary by server

04



IN-APP EXPERIENCE

Obsidian plugin

`agent inside Obsidian`

A community plugin places chat or agent actions directly inside the application.

- + Tight human review loop
- Plugin and provider coupling



The substrate stays the same: Markdown files, links and YAML.

[Official CLI](#)[MCP architecture](#)[Community plugins](#)

Open-source knowledge bases

Three open-source alternatives.

01



Foam

Knowledge management inside VS Code.

- ✓ Plain Markdown files
- ✓ Wikilinks, backlinks & graph
- ✓ CLI with JSON output
- ✓ First-party MCP server

MIT

VS Code

02



SilverBullet

A programmable, self-hosted web application.

- ✓ Markdown-based spaces
- ✓ Bidirectional links
- ✓ Objects, queries & templates
- ✓ HTTP API

MIT

Browser + server

03



Logseq

A privacy-first, block-based outliner.

- ✓ Linked pages & blocks
- ✓ Queries, journals & plugins
- ✓ Markdown or Org files
- ✓ Database-backed mode

AGPL-3.0

Desktop + mobile

[Foam](#)[SilverBullet](#)[Logseq](#)

Karpathy's LLM Wiki · 2026

Knowledge should compound.

 **CONTROL FILE**
AGENTS.md / schema

Defines page types, naming, citations and maintenance rules.



01

RAW**Immutable sources**

Papers, documents, transcripts and data remain the source of truth.

>



02

WIKI**Agent-maintained knowledge**

Linked pages, entities, comparisons, contradictions and evolving synthesis.

>



03

OUTPUTS**Reusable answers**

Useful analyses can be filed back into the wiki instead of disappearing in chat.

While RAG re-derives at query time, an LLM Wiki compiles knowledge once and keeps it current.

[🔗 Karpathy's LLM Wiki](#)

Reading is only half the architecture.

READ PATH

Trusted knowledge → Grounded answer



Ask →



Retrieve →



Inspect →



NEW Answer + cite

WRITE PATH

New material → Trusted knowledge



Draft →



NEW Ground →



NEW Reconcile →



Approve →



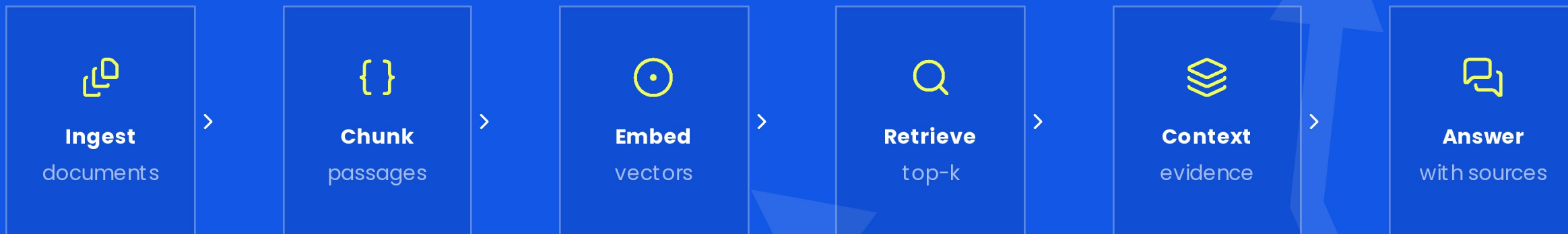
Publish



Compounding knowledge requires **compounding quality controls**: source lineage, contradiction checks, review and rollback.

Document retrieval

Retrieval-Augmented Generation.



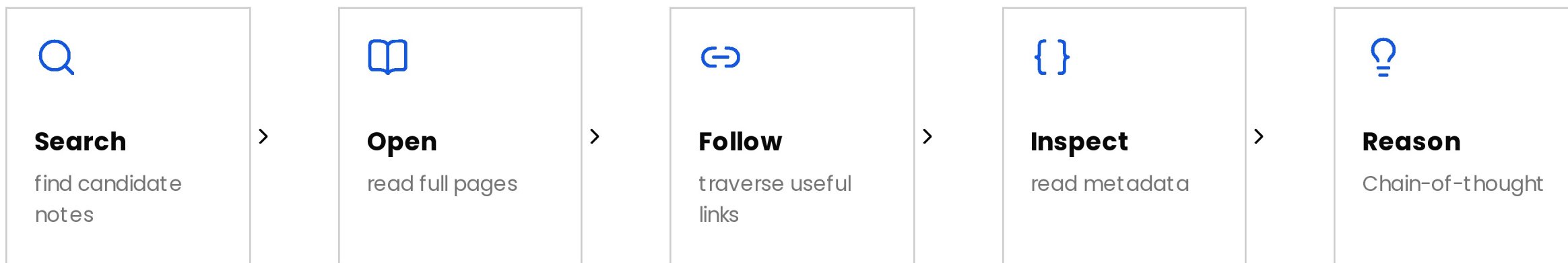
RAG retrieves passages that look relevant, places them in the model context, and asks the model to answer from that evidence.

EVIDENCE BUDGET

Only the retrieved top-k passages reach the model.

Agent access pattern

How can a RAG agent use the wiki?



The agent does not need to memorise the wiki. It needs a **tool contract** for finding, reading and reasoning over it.

Knowledge base access · MCP

Connect through MCP.

ecosystem.graphhub.dobriy.ai

 **European AI Ecosystem**

KNOWLEDGE BASE

[Home](#)
[Trainings](#)
[Organisations](#)
[Technologies](#)
[Infrastructure](#)
[Answers](#)

CONNECTIONS

 MCP access

ONE SET UP FLOW · ANY MCP CLIENT

Connect your IDE to the knowledge base

The endpoint and access token work the same way across MCP-compatible clients.

01



Open & sign in

 **go.dobriy.ai/demo** 

Or import into SilverBullet locally:
[go.dobriy.ai/demo-import](#) 

02



Go to MCP access tab

Copy the connection information and your personal access token.

03



Connect your IDE

Add the MCP connection, then start interacting with your knowledge base.

CONNECT FROM

[Claude Code](#)
[Codex](#)
[Antigravity](#)
[Any MCP client](#)

© 2026 Dobriy AI GmbH



We resume soon

Ten-minute break.

Next: inspect the evidence.

Retrieval is not one problem.

Chunk RAG is strong when one local passage is enough. Graph retrieval becomes valuable when an answer depends on **connected, coordinated or complete evidence**.



LOCAL/FAST RETRIEVAL

Chunk RAG



ONTOLOGICAL REASONING

Graph retrieval

01 Dense retrieval ranks passages independently under a fixed top-k evidence budget.

02 Graph retrieval retrieves connected evidence chains under a matched evidence budget.

03 The experiment locates when exactly the **“reasoning” capability of LLMs breaks down**.

Use document retrieval when...

Where document RAG works well.

01	02	03	04
			
Local facts	Summaries	Semantic discovery	Rich evidence
The answer is contained in one or two passages.	The task is descriptive rather than relational.	The user's wording differs from the source.	Nuance and original prose matter to the answer.

If a good passage already contains the answer, retrieve the passage.

Failure conditions

Where similarity retrieval becomes fragile.

01	02	03	04	05
				
Multi-hop paths	Complete sets	Aggregation	Identity	Changing facts
Event → component → provider and event → system → operator.	Every co-organiser must be checked across every role.	Group contributions by organisation without double-counting.	Cyfronet and Academic Computer Centre Cyfronet AGH must join.	Programmes, tools and event editions change over time.

Similarity can find relevant text. It does not guarantee the **complete relationship structure** required by the question.

The transition

The missing abstraction is explicit structure.



WIKI SENTENCES

Cyfronet co-organised the bootcamp. Athena, Cyfronet's supercomputer, supported the exercises.

Roles described in prose

GRAPH STATEMENTS

Bootcamp — coOrganisedBy —> **Cyfronet**

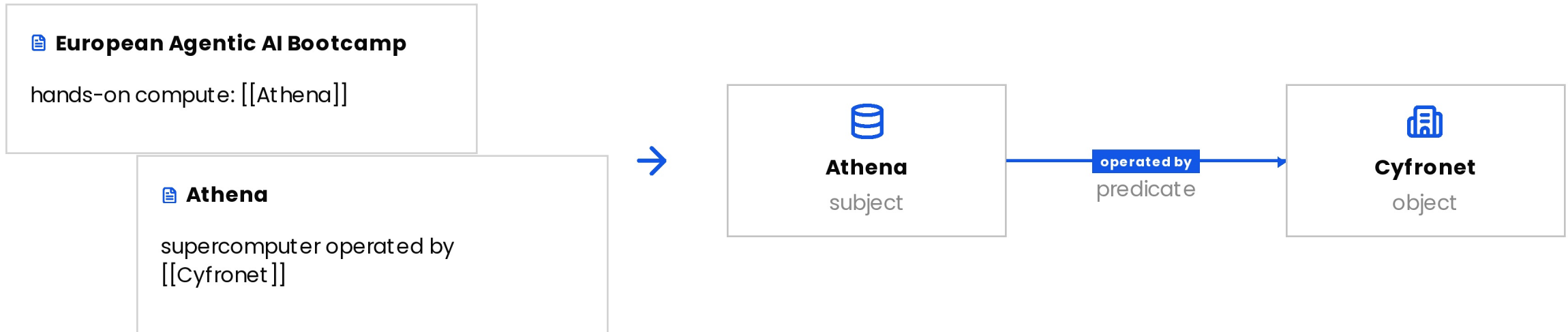
Athena — operatedBy —> **Cyfronet**

Repeated roles queryable as data

The information stays human-readable. The relationship also becomes machine-operable.

Knowledge graphs · 01

A knowledge graph makes links explicit.



A knowledge graph represents entities as nodes and meaningful relationships as typed edges.

RDF triples

A fact has a subject, predicate and object.



```
<athena> <operatedBy> <cyfronet> .
```

Subject

the thing being described

Predicate

the type of relationship

Object

another entity or a value

[W3C RDF concepts](#)

Knowledge graphs · 03

Entities need stable identity.

Athena

Cyfronet's supercomputer

Athena HPC system



CANONICAL IDENTIFIER

<https://ecosystem.graphhub.dobriy.ai/infrastructure/athena-cyfronet>

type: HPCSystem

not: Athena Research Center

✓ Aliases resolve to one entity

✓ Name collisions stay separate

✓ Data from multiple sources can join

Schemas and ontologies

Structure needs shared meaning.

 Event Organisation Technology

coOrganisedBy

taught

providedBy

operatedBy

Simple shared rules

coOrganisedBy connects an event to an organisation.

providedBy connects a technology to its provider.

operatedBy connects infrastructure to its operator.

A schema says what may exist. An ontology makes the intended meaning explicit and reusable.

Trust requires lineage

Facts need provenance.



CLAIM

Athena, Cyfronet's supercomputer in Poland, supported the practical exercises.



AUTHORITATIVE SOURCE

AI:AT bootcamp report



OBSERVED

02 September 2026



STATUS

Verified

source URL

publisher

retrieval date

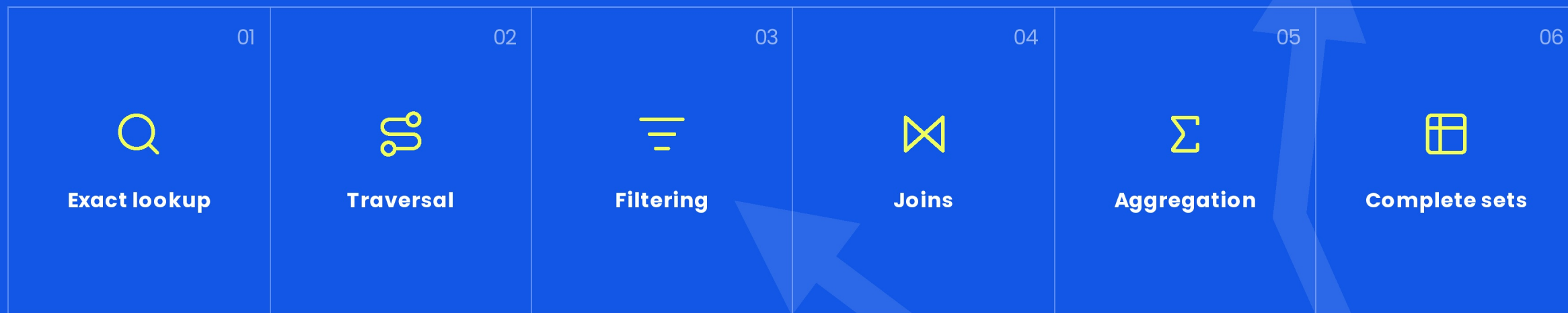
licence

confidence

[Inspect the official AI:AT report](#)

Machine-operable knowledge

What graph structure enables.

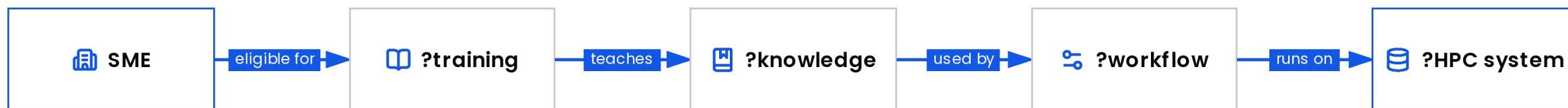


The graph does not replace reasoning. It gives the agent explicit structures it can query before it reasons.

From language to variables

Questions become graph patterns.

Which **training that an SME is eligible for** teaches knowledge used by a workflow running on a **European HPC system**?



Each result must satisfy the complete connected path—not merely match its parts independently.

One readable query

SPARQL queries.

```
SELECT ?stage ?training ?level
  (GROUP_CONCAT(DISTINCT ?resource) AS ?resources)
WHERE {
  VALUES (?stage ?goal) {
    (1 ex:AIAgents) (2 ex:RAG)
    (3 ex:KnowledgeBases) (4 ex:ProductionML) }
  ?training ex:topic ?goal ; ex:level ?level ;
    ex:audience ex:SME .
  OPTIONAL { ?training ex:uses ?resource }
  OPTIONAL { ?training ex:usesInfrastructure ?resource }
}
GROUP BY ?stage ?training ?level
ORDER BY ?stage
```

01 **VALUES** defines the learning stages.

02 **WHERE** filters level, topic and audience.

03 **OPTIONAL** follows tools and compute links.

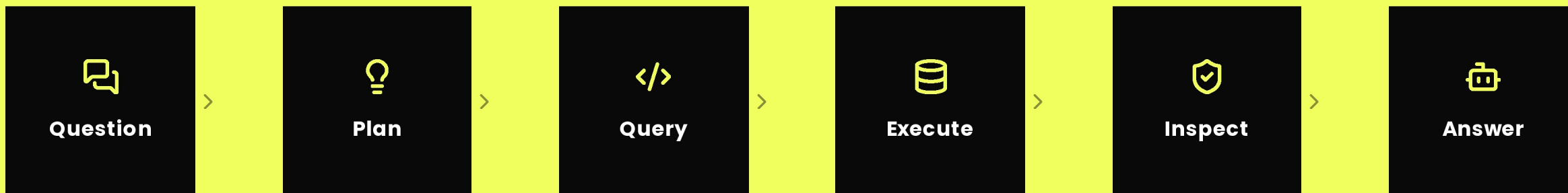
04 **GROUP + ORDER** assemble one usable path.

</> Cypher expresses similar patterns for property graphs. Today we teach one SPARQL query because our example uses RDF.

[W3C SPARQL 1.1](#)

Tool-using agent

The graph becomes an agent tool.

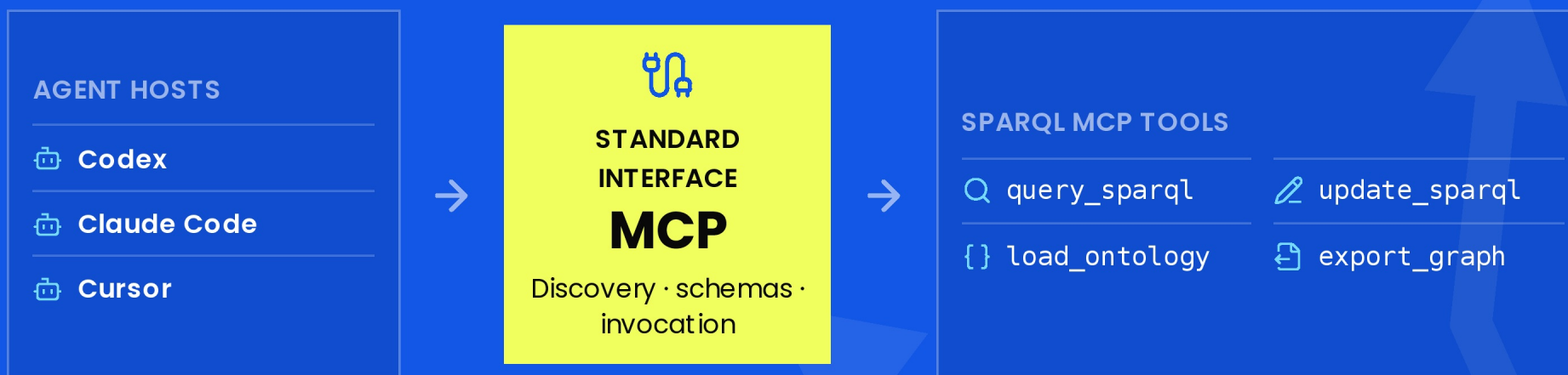


INPUT	TOOL	OUTPUT	POLICY
Natural-language question	SPARQL endpoint	Rows + paths + sources	Timeouts, access and limits

Give the model a narrow, observable capability: **query the graph and return evidence.**

Model Context Protocol

MCP is the connector. Not the memory.



MCP standardises how capabilities are exposed. The server still needs **trusted data, permissions, provenance and operational controls**.

[MCP architecture](#)

Knowledge graph access · MCP

Connect through MCP.

graphhub.dobriy.ai

**European AI Ecosystem**
RDF knowledge graph

SPARQL ENDPOINT

[Query](#)

[Named graphs](#)

[Namespaces](#)

[Ontology](#)

ACCESS

[MCP access](#)

SPARQL ENDPOINT · MCP ACCESS

Connect your IDE to the knowledge graph

Use the same three-step setup flow in every MCP-compatible client.

01



Open & sign in

 go.dobriy.ai/demo-kg 

02



Go to MCP access tab

Copy the MCP server configuration, SPARQL endpoint and your personal access token.

03



Connect your IDE

Add the MCP connection, then query and inspect the knowledge graph.

CONNECT FROM [Claude Code](#) [Codex](#) [Antigravity](#) [Any MCP client](#)

© 2026 Dobriy AI GmbH

Graph RAG · concrete implementation

Private sources become agent-ready context.

01 / INGEST



Add private sources

Import code and documents from a ZIP, repository or workspace.

CODE + DOCS



02 / GRAPH



Build the graph

Parse files, symbols and relationships into a versioned knowledge layer.

STRUCTURED + NAVIGABLE



03 / CONNECT



Serve through MCP

Agents retrieve focused context without receiving the whole source collection.

AGENT-READY

</> Code-aware graph

🔄 Versioned builds

🔍 Focused retrieval

🛡️ Private workspaces

Graphify builds and serves the structured context. **Your preferred agent remains the interface.**

[Graphify AI](#)

Structured + descriptive evidence

The practical architecture is often **hybrid**.



Knowledge graph

Identify entities, traverse relationships, filter and join.

Who? How connected? Which complete set?

+



Documents

Retrieve descriptions, policies, explanations and original wording.

What does it say? Why? With what nuance?



Agent

Combines structure with source-grounded prose.

Graph first to **find repeated roles and complete paths**. Documents next to explain **what each component contributed**.

Architecture selection · 01

Four architecture levels.

-
- 01  **Direct context**
A small, stable bundle placed in the prompt
 - 02  **Wiki / document RAG**
Semantic retrieval over pages and passages
 - 03  **Graph RAG**
Traversal, joins, aggregation and complete sets
 - 04  **Hybrid retrieval**
Graph structure plus descriptive document evidence
-

less structure

more explicit structure

Architecture selection · 02

Choose by question shape.

Requirement	Direct context	Document RAG	Graph RAG	Hybrid
Semantic lookup	•	•	○	•
Multi-hop traversal	○	△	•	•
Aggregation / complete sets	○	○	•	•
Fresh operational values	△	△	•	•
Inspectable provenance	△	•	•	•
Fine-grained access control	○	△	•	•

• natural fit △ possible with h care ○ weak fit

Implementation sequence

From questions to a working agent.

01 Questions Collect representative user questions.	02 Sources Name the authoritative systems.	03 Minimal schema Model only required entities and relations.	04 Identifiers Decide how entities stay stable.
05 Ingestion Preserve freshness and provenance.	06 Query tool Expose narrow, safe operations.	07 Agent Plan, execute and cite.	08 Evaluation Test answers and evidence together.



Do not begin with the biggest ontology. Begin with the **questions whose answers must be reliable.**

Eight-minute design exercise

Design your agent's knowledge architecture.

01 Write one question your agent must answer reliably.

02 Name the authoritative source or system.

03 Mark the update rate: static, periodic or live.

04 Circle relationships, aggregation or audit needs.

05 Choose one of the five architecture levels.

YOUR DESIGN

Question _____

Sources _____

Structure _____

Architecture _____

🕒 5 min design · 3 min debrief

Debrief: What part of your question forced the architecture choice?

Three rules to take away

Questions & discussion.

- 01** Keep evolving facts outside the model.
- 02** The level of knowledge structuring you need depends on the complexity of the question.
- 03** Preserve provenance from source to final answer.

[Workshop page](#)



[GraphHub](#)



[dobriy.ai](#)



Funded by



EuroHPC
Joint Undertaking



Co-funded by
the European Union

 Federal Ministry
Innovation, Mobility
and Infrastructure
Republic of Austria

under discussion with



AI Factory Austria AI:AT has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101253078. The JU receives support from the Horizon Europe Programm of the European Union and Austria (BMIMI / FFG).